

МІО — Споделен слой памет и знание за AI агенти

Продуктова спецификация

Версия 1.2 · Септември 2026

ArtminAI · Ник Настев

1. Какво е МІО

МІО е **слой за памет и знание**, който стои под всички AI инструменти на една организация и им дава общ, постоянен контекст.

Днес всеки AI инструмент помни само себе си. Claude не знае какво сте писали в ChatGPT. Утрешната сесия не знае какво реши вчерашната. Един агент открива проблем, друг го открива отново след седмица.

МІО решава точно това: **едно място за истина, до което всички агенти четат и пишат** — през стандартен протокол (MCP), без промяна в самите инструменти.

Накратко: ако AI агентите са служители, МІО е тяхната обща документация, архив и работна памет.

2. Проблемът, който решава

Без МІО	С МІО
Всеки чат започва от нула	Агентът зарежда контекста за секунди
Знанието живее в главата на един човек	Знанието е записано, търсимо и споделено
Решения се преразглеждат по три пъти	Решенията са записани с причина и дата
Един и същ бъг се дебъгва повторно	Бъговете са в базата със симптом и фикс
Смяна на модел или инструмент = загуба на всичко	Паметта е независима от доставчика
Нов човек в екипа = месеци въвеждане	Нов човек (или агент) пита паметта

3. Ключов принцип: без обвързване с доставчик

МІО не е услуга на конкретна платформа. МІО е архитектура.

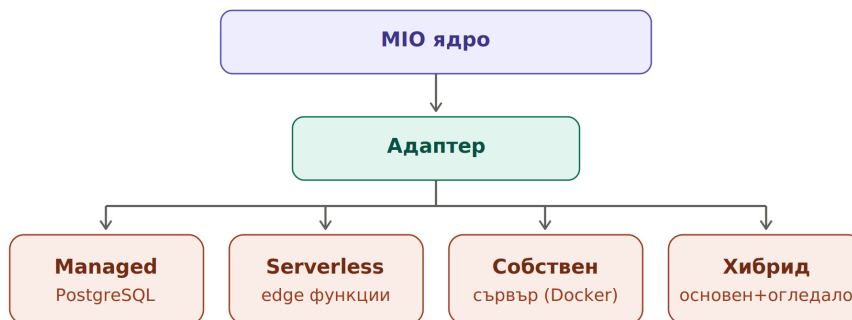
Продуктът се състои от три неща, които са ваши и остават ваши:

- Протоколен слой** — MCP сървър с каталог от инструменти. Това е интерфейсът, който агентите виждат.

2. **Логически слой** — правила за запис, дедупликация, хибридно търсене, граф на същностите, наблюдаемост.

3. **Адаптерен слой** — тънък превод към хранилището отдолу.

Само третият слой знае къде физически лежат данните. Първите два не знаят и не ги интересува.

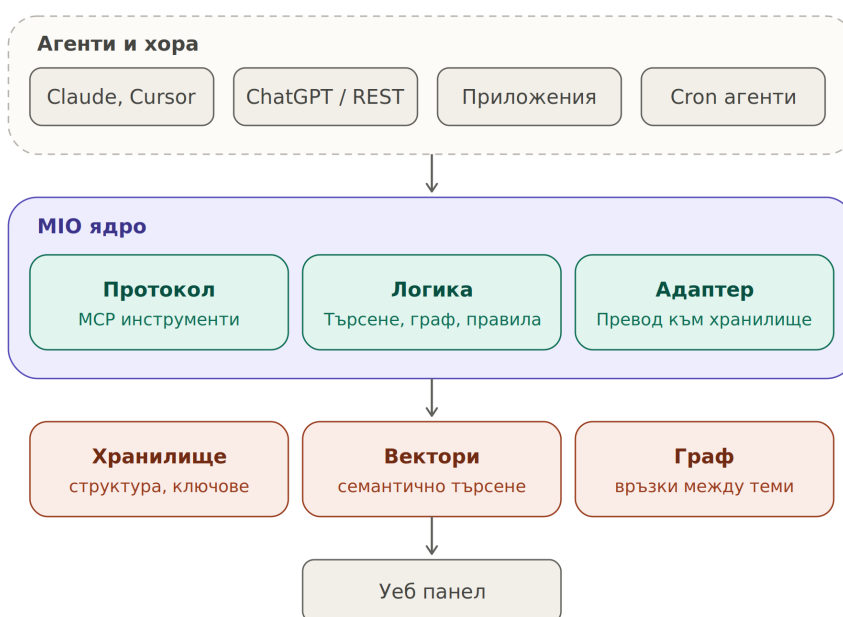


Един и същ MIO — четири различни среди. Сменя се само адаптерът.

Какво значи това на практика:

- Днес работите на managed доставчик — утре мигрирате на собствен сървър. Агентите не забелязват.
- Работите на две места едновременно: основно хранилище + огледало при друг доставчик. При срив системата чете от резервата автоматично.
- Никой доставчик не може да ви вдигне цената или да ви спре, без да имате изход.
- Данните са в стандартен PostgreSQL — не в собствен формат, от който не се излиза.

4. Архитектура



MIO — слоеве. Адаптерът е единственият слой, който знае къде лежат данните.

Принцип: dual-write. Всеки запис влиза едновременно в структурираното хранилище (точни, ключови справки) и във векторния индекс (семантично търсене). Двата пътя се проверяват един срещу друг.

Принцип: dual-read с резерва. При проблем в основния път системата чете от огледалото автоматично, вместо да откаже. Клиентът вижда отговор, не грешка.

5. Възможности

5.1 Памет (Memory)

Ядрото на системата.

Възможност	Описание
Запис	Всяка мисъл, решение, факт или урок се записва с тип, проект, тагове и връзки
Типове	fact, decision, bug, pattern, lesson, preference, note, session
Хибридно търсене	Векторно + текстово търсене, слети с RRF — намира и по смисъл, и по точна дума
Извикване по ключ	Прецизно вадене на конкретен запис без търсене
Обновяване	Записът се коригира на място, историята се пази
Забравяне	Меко изтриване (tombstone) — обратимо, с одит следа
Дедупликация	Семантична — един и същ факт не влиза два пъти под различни думи
Сесийна памет	Автоматично резюме на работната сесия + извлечени ключови факти
Многоагентност	Всеки агент има идентичност; вижда общото и своето

5.2 Граф на същностите (Entity Graph)

Паметта отговаря не само на „*какво съм казвал за X*“, а и на „*какво е свързано с X*“.

- Автоматично разпознаване на същности: проекти, платформи, хора, клиенти, инструменти, теми
- Детерминистично извличане (речник + правила) — без разход за модел на всеки запис
- Връзки с тегло, сортирани по сила
- Отговаря на въпроси от типа: „*кои проекти ползват този инструмент*“, „*какво зависи от тази база*“, „*кое е на този клиент*“
- Графът се вгражда автоматично в резултатите от търсене — не изисква нов начин на питане

Референтен показател: при реален масив от ~12 000 записа графът произвежда **11 859 връзки**, с таван срещу размиване и покритие, следено ежеседмично.

5.3 Проектна памет

Всеки проект получава собствено досие с четири задължителни части:

1. **Snapshot** — къде е проектът в момента, фаза, напредък
2. **Todo** — единственият източник за „какво остава“, всеки ред с критерий за проверка
3. **Решения** — какво е решено, кога и защо
4. **Бъгове и уроци** — какво се е случило и как е поправено

Агентът започва сесия с реалното състояние, не с предположение.

5.4 База знание и RAG

- **Качване на документи** — разбиване на смислови парчета, вграждане, индексирание
- **Колекции** — знанието се разделя по теми (нормативна база, продуктова документация, вътрешни правила)
- **Семантично търсене** в чънковете с опция за препоредждане (reranking) от модел
- **Категории и типове** — лично / работно / техническо; бележка / документ / публикация
- **Автоматична синхронизация** — външни документационни страници се следят и при промяна се пре-индексират

5.5 Планер и задачи

- Задачи с проект, приоритет, статус и изпълнител
- Достъпни едновременно за човек (панел) и за агент (MCP)
- Агентът може да отваря, чете и затваря задачи в рамките на работата си
- Връзка задача ↔ памет: решението, породило задачата, е на един клик

5.6 Автоматичен контекст

Един извик в началото на сесията връща най-важното по темата: решения, бъгове, шаблони, задачи, знание. Оптимизиран за токени (~2000), за да не яде контекстния прозорец.

Ефект: **агент, който знае историята, без човек да я преразказва.**

5.7 Уеб панел (Dashboard)

За хората в екипа, които не работят през агент:

- Преглед и търсене в цялата памет
- Бележки — бърз вход от телефон или браузър
- Проекти и статуси
- Информационен / watch поток (по избор)
- **Мониторинг** — състояние на системата, капацитет, разход, здравни индикатори

- Достъпен на собствен домейн на клиента

5.8 Наблюдаемост и здраве

Проектиран така, че **тихата повреда да е невъзможна**.

- Здравни сигнали за всеки критичен път (вграждане, запис, огледало, квоти)
- Показва **ефективния остатък**, не подвеждащ общ баланс
- Три състояния: здраво / ограничено / непотвърдено — „не знам“ никога не се представя като „наред е“
- Планирани проверки за покритие, дрейф и растеж
- Известия само при праг — тишината е част от дизайна, не липса на следене
- Пълна одит следа: кой агент, кога, какво е записал

5.9 Устойчивост и преносимост

- **Двупосочно огледало** между основното хранилище и резервното — при различни доставчици, ако желаете
- **Автоматичен fallback** при недостъпност на основния път
- **Одит на дрейфа** — инструмент, който сравнява двете страни и попълва липсващото
- **Пълен бекъп** — схема, данни, скрипт за възстановяване, тестова процедура
- **Износ във всеки момент** — стандартен SQL дъмп плюс векторите; нищо не е заключено
- Тайните се съхраняват отделно от архива на данните

6. MIO в организация с много служители

Стойността на MIO расте с броя хора, които го ползват. При един човек той е удобство. При тридесет — той е разликата между организация, която помни, и организация, която всеки път започва отначало.

6.1 Защо мащабът работи в полза на системата

Знанието в един екип не се сумира линейно, а се пресича. Тридесет души произвеждат стотици припокриващи се въпроса: *кой клиент какво иска, защо е избрано това решение, къде е онзи документ, какво се случило миналия път*. Днес отговорите се търсят с питане на колега — най-скъпият вид търсене, защото спира двама души.

MIO превръща това в заявка към паметта. Колкото повече хора пишат в нея, толкова по-често следващият намира готов отговор.

Брой хора	Какво се променя
1-5	Лична памет, приемственост между сесии
5-20	Общ контекст по проекти, край на повторните обяснения
20-100	Организационна памет — въвеждането на нов човек пада от месеци на дни
100+	Отделите се разделят на пространства, споделеното знание остава общо

6.2 Пространства: общо, екипно, лично

Паметта се разделя на три нива:

- **Организационно** — правила, процеси, продуктово знание, нормативна база. Чете се от всички.
- **Екипно / проектно** — досието на проекта: решения, бѣгове, задачи, статус. Чете се от екипа и от агентите, работещи по него.
- **Лично** — бележки, чернови, работни хипотези на отделния служител. Не изтича в общото, освен ако човекът не го публикува.

Разделението е на ниво маршрут и права, не на ниво „договорка да не гледаме“. Един служител физически не вижда чуждото лично пространство.

6.3 Роли и права

Роля	Може
Читател	Търси и чете общото и екипното си пространство
Сътрудник	Плюс запис в своето екипно и лично пространство
Отговорник на проект	Плюс редакция и затваряне на записи в проекта си
Администратор	Права, пространства, ключове, интеграции, одит
Агент	Права, наследени от човека или екипа, от чието име работи

Ключово: **агентът никога няма повече права от човека, който го е пуснал**. Ако служителят не вижда финансовото пространство, неговият агент също не го вижда.

Статус: пространствата, изолацията и ключовете с обхват работят днес. Ролевият модел с права по роля е проектиран и е следващата фаза от разработката — при внедряване в организация се доуточнява спрямо реалната структура на клиента. Казвам го изрично, за да не се купува нещо, което още не е пуснато.

6.4 Въвеждане на нов служител

Новият човек получава достъп до общото пространство и до екипа си, и от първия ден задава въпроси на паметта вместо на колегите:

- *Защо ползваме този доставчик, а не другия?*
- *Кой е отговорникът по клиента X и какво е договорено?*

- Как се пуска *production* деплой при нас?
- Какво се случило последния път, когато пипахме това?

Отговорите идват с дата, автор и източник. Наставникът се включва там, където паметта е празна — а празнотата се вижда и се запълва.

6.5 Отчетност и одит

- Всеки запис носи **автор, агент, време и източник** — знае се кой какво е внесъл
- Меко изтриване с проследима следа — нищо не изчезва безшумно
- Отчет за покритието: кои проекти имат актуално досие и кои са изостанали
- Отчет за приноса: кои екипи хранят паметта и кои само четат — управленски сигнал, не наказание

6.6 Какво расте с броя хора (и какво трябва да се планира)

Честно за разходите, защото при организация те не са пренебрежими:

Расте	Как се управлява
Брой записи и вектори	Векторите са ~80% от обема; размерността се сваля без ре-вграждане
Разход за вграждане	Ключ с таван на организация или на екип, видим в панела
Трафик и заявки	Кеширане, пакетни операции, измерване по извикване, не по усет
Шум в паметта	Семантична дедупликация + правила за това какво заслужава запис
Права и пространства	Администраторски панел; при 100+ души — синхронизация с директорията на клиента

Размерът на базата при ~12 000 записа е около 400 MB. Организация от 50 души, която пише дисциплинирано, стига до този обем за няколко месеца — тоест говорим за десетки гигабайти на година, не за терабайти.

6.7 Условието, което не се заобикаля

МІО не е инструмент, който се инсталира и работи. При организация от много хора успехът зависи от едно нещо:

Трябва да е ясно кога се пише в паметта — и това трябва да е задължение, не добра воля.

Практически това значи няколко правила, вградени в работния процес: край на проект = записано досие; взето решение = записана причина; поправен бъг = записан симптом и фикс. Агентите поемат голяма част от това автоматично, но правилото трябва да съществува.

Организации, които въведат МІО без това правило, получават празна база и разочарование. Затова фаза 3 от плана за въвеждане е точно за работните навици, а не за технология.

7. Интеграции

Готови днес:

Канал	Как
Claude.ai	MCP connector
Claude Code / CLI	MCP
Cursor, VS Code, Copilot CLI	MCP
ChatGPT / произволен модел	REST API
Google Workspace (Gmail, Calendar, Drive)	конектор
Microsoft 365	конектор
Telegram / e-mail	известия
Гласов интерфейс	собствен транспорт
Cron / планирани агенти	REST
Собствени приложения	REST + ключ с обхват

Принцип: MIO не заменя инструментите ви. Той се закача под тях.

8. Сигурност

Мярка	Реализация
Изоляция на данните	Всеки клиент — собствен префикс за маршрути и собствени ключове
Row Level Security	Включен на всички таблици с данни
Публичен достъп	Изрично отнет за паметта и графа на същностите
Ключове с обхват	Отделни ключове за четене, запис и админ операции
Тайни	Отделно хранилище, никога в архиви или логове
Защита от изтичане	Автоматични проверки при всеки деплой
Одит	Всеки запис носи автор, агент, време и източник
Разход за модели	Вграждането минава през ключ с таван — контролиран и видим разход

9. Технически изисквания

MIO не изисква конкретна платформа. Изисква **четири способности**, които се покриват от много различни доставчици:

Способност	Какво трябва	Примерни варианти
Структурирано хранилище	PostgreSQL 14+	Managed PostgreSQL, self-hosted, всеки облак
Векторен индекс	pgvector с HNSW	Същата база, или отделен векторен store
Изпълнение на код	Serverless функции или Node процес	Edge функции, контейнер, VPS
Планирани задачи	Cron	pg_cron, системен cron, планировчик на доставчика

Ядро: TypeScript / Node.js, функционален стил, без зависимост от SDK на конкретен облак.

Вграждане: модел от типа `text-embedding-3-small` (Matryoshka — размерността се сваля без ре-вграждане). Доставчикът се сменя с конфигурация; поддържа се и локален модел.

Панел: статичен фронтенд — работи от всеки CDN, обектно хранилище или собствен nginx.

Доказани среди

Ядрото е пуснато в производство върху три различни комбинации:

- Managed PostgreSQL + edge функции + CDN хостинг
- Serverless платформа с интегрирана база и векторно търсене
- Локален Docker (PostgreSQL + Node) на собствена машина

Миграцията между тях е смяна на конфигурация и пренос на данните — не пренаписване.

Показатели от реален deployment

Метрика	Стойност
Записи в паметта	~12 300
Векторизирани	100% от нормалния път
Връзки в графа	11 859
Отговор при търсене (облачна база)	100–160 ms
Отговор при локална инстанция	~1 ms
Пълен контекст в началото на сесия	~2–4 s
Уникалност на ключовете	12 126 / 12 126, нула дубликати
Размер на базата при 12k записа	~400 MB (80% от него вектори)

10. Модели на внедряване

А. Управлявана инстанция

Хостваме и поддържаме ние, върху инфраструктура по наш избор. Клиентът получава собствена изолирана среда, ключове и панел.

За: малки екипи, бърз старт, без вътрешен DevOps.

В. Ваша инфраструктура, наша поддръжка

Инсталираме MIO върху акаунт и доставчик на клиента — какъвто и да е той. Сметките и данните са на клиента; ние поддържаме слоя.

За: компании с изградена облачна практика.

С. Локално / on-premise

Целият слой работи в Docker в мрежата на клиента. Данните не я напускат.

За: регулирани сектори, чувствителни данни.

Забележка: вграждането може да остане външно или да мине на локален модел — по избор.

Д. Хибрид

Основно хранилище на едно място, огледало на друго. Автоматична резерва при срив на едното.

За: критични натоварвания и екипи, които не искат единична точка на отказ.

Преминаването между модели $A \rightarrow B \rightarrow C$ е предвидено от самото начало. Не е миграционен проект, а конфигурация плюс пренос на данни.

11. Как протича въвеждането

Фаза 1 — Основа

Избор на среда, инстанция, ключове, MCP връзка към инструментите на клиента, панел. Първи записи.

Фаза 2 — Пренасяне на знание

Качване на съществуваща документация, проектни досиета, вътрешни правила. Настройка на колекции и категории.

Фаза 3 — Работни навици

Правила кога агентът чете и кога пише. Шаблони за сесиен запис. Обучение на екипа — паметта работи само ако в нея се пише.

Фаза 4 — Автоматизация

Планирани брифинги, автоматична синхронизация на документация, известия, граф на същностите.

Фаза 5 — Наблюдение и настройка (постоянно)

Здравни сигнали, покритие, прагове, оптимизация на разхода.

12. Какво MIO не е

Честно, за да няма изненади:

- **Не е модел.** MIO не мисли и не генерира. Той помни и подава. Моделът си остава ваш избор.
- **Не е заместител на база данни за приложението.** Той е памет за агенти, не транзакционна система.
- **Не работи сам.** Ако агентите не пишат в него, той остава празен. Дисциплината по записа е част от внедряването, не опция.
- **Семантичното търсене не е точно търсене.** За гарантирано точни справки се ползва ключ или структуриран филтър.
- **Качеството на графа зависи от речника на домейна.** Първоначалната настройка е задължителна част от работата.
- **Преносимостта не е безплатна.** Смяната на среда е предвидена и документирана, но иска планиран прозорец и повторно издаване на ключове.

13. Пътна карта

Приоритет	Какво
Следващо	Филтриране по тагове в RAG търсенето · оптимизация на размерността на векторите · директно качване на знание без междинен слой
След това	Модел за откриване на непокрита същности (фаза 2 на графа) · декларираните връзки между записи · разширени права по роля
По-нататък	Локален модел за вграждане · самообслужване при създаване на инстанция · публично API за партньорски интеграции

14. Следваща стъпка

Най-бързият начин да се прецени MIO е **демонстрация върху ваши данни**: качваме един реален проект или комплект документация в тестова инстанция и показваме разликата — един и същ въпрос, зададен на агент със и без MIO.

Средата за демото избирате вие. Ако нямате предпочитание, пускаме я на наша и после я местим.

Пишете ми и ще уговорим обхвата.

Контакт

Ник Настев · ArtminAI
Варна, България